

THREE-DIMENSIONAL SIMULATION OF NONWOVEN FABRICS USING A GREEDY APPROXIMATION OF THE DISTRIBUTION OF FIBER DIRECTIONS

SIMONE GRAMSCH, MAX KONTAK, AND VOLKER MICHEL

ABSTRACT. An elementary algorithm is used to simulate the industrial production of a fiber of a 3-dimensional nonwoven fabric. The algorithm simulates the fiber as a polyline where the direction of each segment is stochastically drawn based on a given probability density function (PDF) on the unit sphere. This PDF is obtained from data of directions of fiber fragments which originate from computer tomography scans of a real non-woven fabric. However, the simulation algorithm requires numerous evaluations of the PDF. Since the established technique of a kernel density estimator leads to very high computational costs, a novel greedy algorithm for estimating a sparse representation of the PDF is introduced. Numerical tests for a synthetic and a real example are presented. In a realistic scenario, the introduced sparsity ansatz leads to a reduction of the computation time for 100 fibers from nearly 40 days to 41 minutes.

1. INTRODUCTION

According to EDANA, the European Disposables and Nonwovens Association, a nonwoven is defined as “...a sheet of fibres, continuous filaments, or chopped yarns of any nature or origin, that have been formed into a web by any means, and bonded together by any means, with the exception of weaving or knitting.” (cited in [30]). Their impact on our daily live is large due to their versatile properties. Nonwovens can be absorbent, antistatic, breathable, conductive or non-conductive, elastic, flame resistant, impermeable or permeable, smooth, and stiff to name but a few. These properties are achieved by combining raw materials with specific production processes.

There are three main nonwoven manufacturing processes: dry-lay processes, wet-lay processes, and extrusion processes. In the following, we concentrate on a typical extrusion process, the so-called spunbond process. A sketch of this process can be found in Figure 1. In this production process, a polymer melt is extruded through spinnerets. The fibers evolving from the spinnerets are cooled and stretched by

(SG) DEPARTMENT TRANSPORT PROCESSES, FRAUNHOFER ITWM, FRAUNHOFER-PLATZ 1, 67663 KAISERSLAUTERN, GERMANY

(MK,VM) GEOMATHEMATICS GROUP, DEPARTMENT OF MATHEMATICS, UNIVERSITY OF SIEGEN, WALTER-FLEX-STR. 3, 57068 SIEGEN, GERMANY

E-mail addresses: simone.gramsch@itwm.fraunhofer.de, kontak@mathematik.uni-siegen.de, michel@mathematik.uni-siegen.de.

2010 *Mathematics Subject Classification.* 37M05, 65C50, 65D15.

Key words and phrases. Abel-Poisson kernel, big data, density estimation, directional statistics, greedy algorithm, nonwoven fabric.

MK and VM gratefully acknowledge the financial support by the School of Science and Technology of the University of Siegen, Germany.

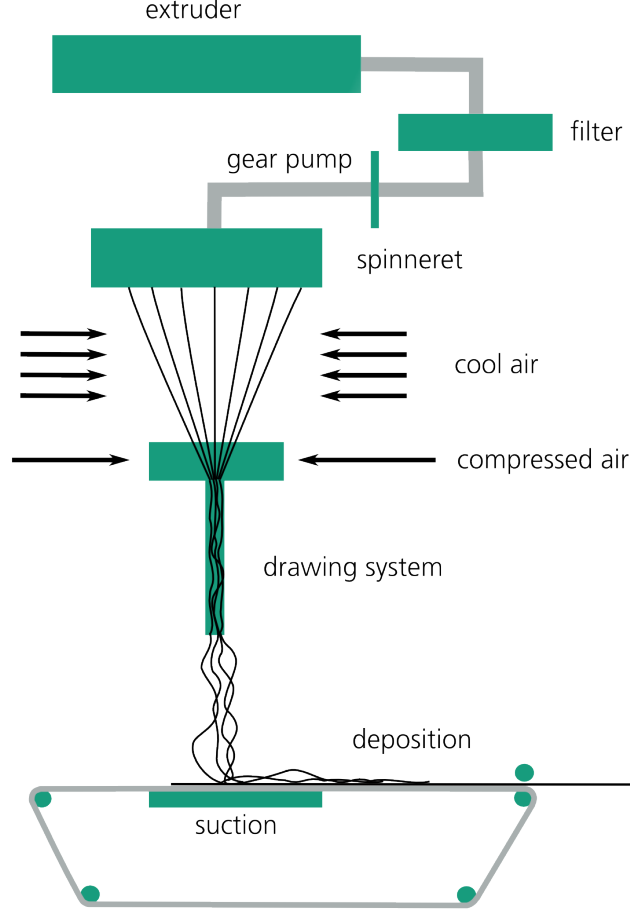


FIGURE 1. Sketch of a nonwoven production process (spunbond)

air. Additionally, they are swirled around by turbulent air streams until they are deposited on a perforated conveyor belt. By suction, they are fixed on the belt to form a random web. In order to produce a nonwoven fabric, additional process steps like bonding and finishing have to be implemented. For further details we refer to [1].

The advantage of nonwovens to be specially designed to have specific properties also has a drawback. The market of nonwovens demands more and more customized products, so the development cycles are shortened accordingly. Hence, simulating nonwoven production processes is a mathematical key technology that enables engineers to design the processes with respect to customer-specific needs.

Several papers have already dealt with the mathematical modeling of this specific production process. One model that is based on the description of the physical process itself can be found in [25, 44], which is also used in the simulation software

FIDYST (fiber dynamics simulation tool, cf. [19]). In principle, one has to deal with a two-way coupling problem of the aerodynamic forces and the fiber dynamics, which is not solvable to industrial scales due to the required resolution of the mesh (cf. [31]). Therefore, the model only incorporates a one-way coupling by using an air drag model. Up to now, this approach neglects the interaction of one filament with itself or other filaments on the conveyor belt that have already been laid down. Therefore, the simulated laydown of filaments on the conveyor belt is two-dimensional.

Due to the high computational effort needed for simulating this model, a real nonwoven consisting of thousands of fibers cannot be simulated in a reasonable amount of time. Hence, in [21, 25] a different approach was developed. The two-dimensional fiber laydown is described by a surrogate model based on a stochastic differential equation. The parameters that arise in this equation can be estimated from the FIDYST simulation of one single fiber. Then, a considerable number of fibers representing the microstructure of a real nonwoven can be computed in a short time using the stochastic differential equation model.

On the one hand, a two-dimensional simulated fabric still represents typical nonwovens, which are very flat, in an excellent way. On the other hand, several properties of nonwovens originate in the fact that the fibers are lying on top of each other, which cannot be achieved by a two-dimensional simulation. Thus, a three-dimensional model is desirable.

For that reason, in [26] the concept of the stochastic surrogate model was extended to a three-dimensional setting. Although possessing good theoretical properties, this model lacks the possibility of estimating the parameters directly using a FIDYST simulation, since the output of the simulation is only two-dimensional. Parameter estimation for the three-dimensional model by combining two-dimensional FIDYST data and three-dimensional data obtained by CT scans has been presented in [20] under the restriction that a non-moving belt or negligible small belt speeds are considered.

In section 2 of this paper, we will present an algorithm for the three-dimensional simulation of nonwovens based on the analysis of fiber directions inside a real nonwoven. Our algorithm generates a polyline by sampling from a probability density function, which has to be estimated first. This estimation is based on a data set generated by image analysis techniques. Here, we use data obtained by 3D-microtomography with a resolution of $1\text{ }\mu\text{m}$ to $70\text{ }\mu\text{m}$ voxel side length and a maximal size of the sample of 1 mm^3 to 10 mm^3 . A reduced version of the data set generated by the computer tomography (CT) scan is shown in Figure 2. Note that in a post-processing step, the raw CT data are translated to filament directions. Since directions in the three-dimensional space can be equivalently described as points on the sphere, every data point on the sphere in Figure 2 corresponds to the filament direction in one single voxel. For a more detailed description of the data set that is used, consult section 5 of this paper. The resulting number of occurring voxels is extremely large due to the high resolution. Hence, efficient strategies for analyzing the data must be developed.

In section 3, we will point out that, in principle, a standard technique of nonparametric statistics for the estimation of probability density functions, namely kernel density estimators, can be used to implement the simulation algorithm. However, it is not efficient to use these estimators for very large data sets as in our case.

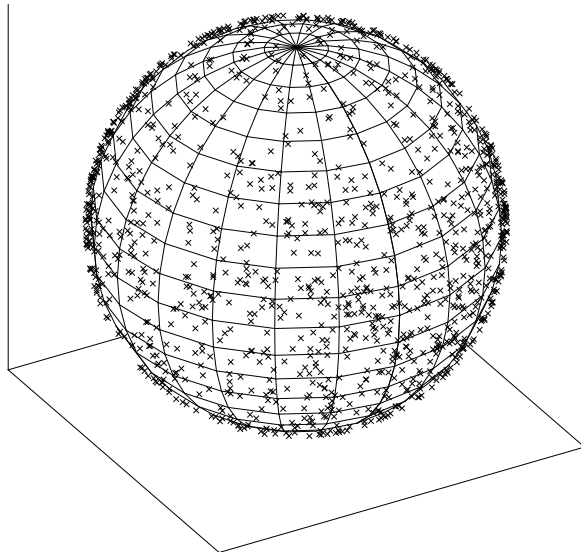


FIGURE 2. 2000 out of 9 600 558 points from a data set of directions in a real nonwoven fabric

In particular, the fiber simulation requires numerous evaluations of the estimated density.

Therefore, we will derive a greedy algorithm for the estimation of probability density functions in section 4.

In section 5, we will first study the convergence of the greedy algorithm empirically and will then show that the efficiency of our simulation algorithm can be enhanced extremely by using the greedy algorithm to estimate the distribution of fiber directions. We will furthermore present first results about the validation of fiber laydown simulation methods like FIDYST, which can be performed using the presented greedy algorithm.

2. MATHEMATICAL SETTING AND THE ALGORITHM

No matter if mathematically simulated fibers are considered or if CT-based data of nonwoven fabrics are available, in each case we are confronted with a set of $N \in \mathbb{N}$ directions in the plane (2D case) or the space (3D case). We will model these data X_1, \dots, X_N as independent and identically distributed realizations of an \mathbb{S}^{d-1} -valued random variable X , where $d = 2$ or $d = 3$. By $\mathbb{S}^{d-1} := \{ \xi \in \mathbb{R}^d \mid |\xi| = 1 \}$, we denote the unit sphere in \mathbb{R}^d . Moreover, the distribution of X shall be absolutely continuous with respect to the surface measure ω on \mathbb{S}^{d-1} , that is, there exists a probability density function (PDF) $f : \mathbb{S}^{d-1} \rightarrow [0, \infty)$, such that for every subset $A \subseteq \mathbb{S}^{d-1}$ we have

$$\mathbb{P}(X \in A) = \int_A f(\xi) d\omega(\xi),$$

where $\mathbb{P}(X \in A)$ denotes the probability that the random variable X attains values in the subset A . Note that we do not impose further assumptions on the PDF,

especially not on any parametrization, such that the estimation of the PDF is a problem in the field of nonparametric statistics.

In the following, we propose a data-driven algorithm for the simulation of non-woven fabrics. The concept of the algorithm is to start at an arbitrary point in \mathbb{R}^d and to choose the next point of a fiber by sampling a direction from the estimated PDF and moving into this direction by a given step width. By construction, the resulting fiber approximately has the same distribution of directions as it occurs inside the given data set used for estimating the PDF. This idea translates to an algorithm as follows.

Algorithm 1. *Let an estimate \hat{f} of the PDF f as well as a discretization parameter $s \in (0, \infty)$ be given.*

Generate a discretization $(Z_j)_{j=0,1,\dots} \subseteq \mathbb{R}^d$ of a fiber by the following iteration:

- (1) Set $j := 0$ and choose an initial point $Z_0 \in \mathbb{R}^d$.*
- (2) Sample a direction $Y_{j+1} \in \mathbb{S}^{d-1}$ from the estimated PDF \hat{f} .*
- (3) Set $Z_{j+1} := Z_j + sY_{j+1}$.*
- (4) If the desired number of points is reached: stop.*

Otherwise: increase j by 1 and return to step 2.

Note the resemblance of the algorithm to the stochastic time discrete approximation of stochastic differential equations, in particular the Euler-Maruyama method, where Gaussian pseudo-random numbers are used in a similar way if the stochastic differential equation involves Brownian motion (cf. [28], chapter 9). However, it is not trivial if there is an interpretation of our algorithm as an Euler-Maruyama scheme for a certain stochastic differential equation and even if it was, it is completely unclear what such a stochastic equation or a stochastic process solving this equation would look like.

Note that the approach for the fiber simulation presented here differs essentially from other methods such as the one described in [25]. For example, we do not solve a given stochastic differential equation which includes an explicit term for the belt movement as it is done in other cases. However, in our numerical experiments in section 5, we will use a PDF which is determined from data of a real nonwoven and, thus, incorporates the non-uniform distribution of the fiber directions with a dominance of the direction of the belt movement. In this respect, the belt movement implicitly becomes an ingredient in our Algorithm 1.

A standard tool to solve the problem of estimation of PDFs is a so-called kernel density estimator, which is briefly discussed in the following section.

3. ANALYSIS OF DIRECTIONS WITH KERNEL DENSITY ESTIMATORS

Since their introduction in [36], kernel density estimators (KDEs) have been a tool that has often been used for the estimation of PDFs based on given data sets X_1, \dots, X_N . Being introduced on the real line, extensions to the multivariate case (see [5]) were straight-forward, which made this method superior to older methods like, for example, the finite difference approximation in [38]. In [22], KDEs were transferred to the case of spherical data $X_1, \dots, X_N \in \mathbb{S}^{d-1}$ for $d \geq 2$, which is of particular importance for the presented application and is, thus, the foundation of the following considerations.

In analogy to the latter reference, in this paper a *kernel* is the member of a family of integrable functions $K_\lambda : [-1, 1] \rightarrow [0, \infty)$, which depend on a so-called

concentration parameter $\lambda \in (0, \infty)$, and have the following properties, that have also been considered in [14] to obtain approximate identities on the sphere:

(a) normalization:

$$\int_{-1}^1 K_\lambda(t) dt = 1, \quad (1)$$

(b) concentration at 1: for every $c \in (-1, 1)$, we have

$$\lim_{\lambda \rightarrow \infty} \int_c^1 K_\lambda(t) dt = 1. \quad (2)$$

The latter is different from the definition of kernels on \mathbb{R} , where a concentration at 0 is required, because for $\xi, \eta \in \mathbb{S}^{d-1}$ the propositions $|\xi - \eta| = 0$ and $\xi \cdot \eta = 1$ are equivalent and the kernel will always be used in the form $K_\lambda(\xi \cdot \eta)$. Here, by $x \cdot y$ we denote the Euclidean inner product of two vectors $x, y \in \mathbb{R}^d$. Under certain further assumptions on the family of kernels it was proved in [22] that the associated KDE, which is defined as

$$\hat{f}(\xi) := \frac{1}{N} \sum_{n=1}^N K_\lambda(X_n \cdot \xi), \quad (3)$$

is an asymptotically unbiased and consistent estimator of the PDF f if λ is chosen properly in dependence of N .

We use the d -dimensional Abel-Poisson kernel $Q_h^{(d)} : [-1, 1] \rightarrow (0, \infty)$,

$$Q_h^{(d)}(t) := \frac{1}{\omega(\mathbb{S}^{d-1})} \frac{1 - h^2}{(1 + h^2 - 2ht)^{d/2}} \quad (4)$$

for $d = 3$ in the following to obtain a KDE for a given CT data set, which consists of points on \mathbb{S}^2 . The term $\omega(\mathbb{S}^{d-1})$ denotes the surface of the unit sphere \mathbb{S}^{d-1} in \mathbb{R}^d , which can be explicitly computed as

$$\omega(\mathbb{S}^{d-1}) = \frac{2\pi^{d/2}}{\Gamma(\frac{d}{2})},$$

which reduces to the well-known results $\omega(\mathbb{S}^1) = 2\pi$ and $\omega(\mathbb{S}^2) = 4\pi$ in the two- and the three-dimensional case, respectively.

Note that the Abel-Poisson kernel fulfills the conditions (1) and (2) if we set

$$K_\lambda(t) := Q_{1-1/\lambda}^{(d)}(t).$$

For further properties of the Abel-Poisson kernel in the three-dimensional case, see for example [32, Section 5.1].

In Figure 3, the KDE with respect to the CT data set is depicted on \mathbb{S}^2 . The parameter $h = 0.9$ is more or less arbitrarily chosen, since a rigorous method for parameter selection is not known, whereas heuristics, which are available, could of course be used to determine the parameter (for an overview on this subject, see [24] and the references therein). The figure shows that there exists a dominant direction of the fibers, which corresponds to the direction in which the belt moves in the production process.

Concerning Algorithm 1, it turns out in numerical experiments that the use of a KDE as estimator of the PDF inside this method is not practicable from the computational point of view. The crucial point is step 2 of the algorithm, where sampling from the estimated PDF \hat{f} is needed.

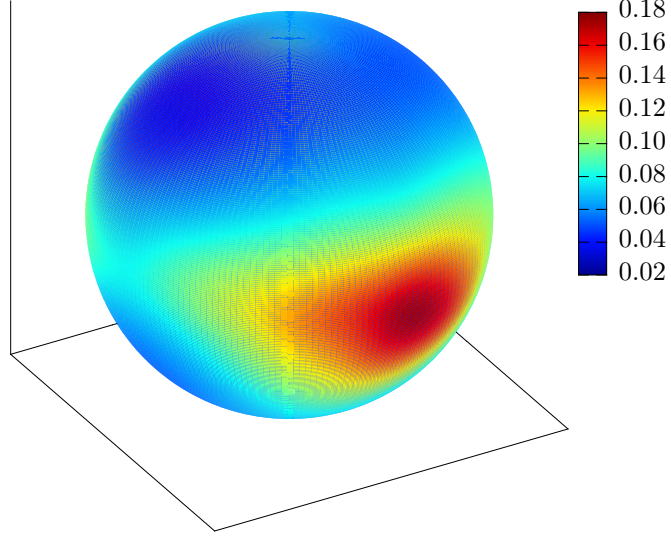


FIGURE 3. Plot of the KDE obtained from the data set where the Abel-Poisson kernel with parameter $h = 0.9$ was used as the kernel in (3).

In our experiments—where we worked with a real CT data set on \mathbb{S}^2 with $N = 9\,600\,558$ directions—we used the well-known acceptance-rejection method (see, e. g., [7, Section II.3]) for sampling. Simpler sampling methods like inversion sampling (see, e. g., [7, Section II.2]) are unfortunately not applicable on spherical domains.

The acceptance-rejection method is based on the following idea (cf. [7, Theorem II.3.1]): If (X, U) is a $\mathbb{S}^{d-1} \times \mathbb{R}$ -valued random variable, which is uniformly distributed on the set $A = \{(\eta, t) \mid \eta \in \mathbb{S}^{d-1}, 0 \leq t \leq f(\eta)\}$, then X has the density f . In practice, a sample (ξ, u) of (X, U) can be obtained by first generating a sample $(\tilde{\xi}, \tilde{u})$ from a uniform distribution on $\tilde{A} = \{(\eta, t) \mid \eta \in \mathbb{S}^{d-1}, 0 \leq t \leq C\}$ for some upper bound C of f . If $\tilde{u} \leq f(\tilde{\xi})$, the sample is *accepted* and $(\xi, u) := (\tilde{\xi}, \tilde{u})$ is taken as a sample from the uniform distribution on A . On the other hand, if $\tilde{u} > f(\tilde{\xi})$, the sample is *rejected* and the generated pair is not taken into account for the samples from the uniform distribution on A . Since methods for generating uniform distributions both on \mathbb{S}^{d-1} and on the interval $[0, C]$ are well-known and implemented in well-established software like the GNU Scientific Library [18], samples from a uniform distribution on $\tilde{A} = \mathbb{S}^{d-1} \times [0, C]$ can easily be generated both in theory and in practice.

Note that the efficiency of the acceptance-rejection method depends heavily on the quality of the upper bound C of the PDF. When an estimation of the PDF is given by a KDE, a simple and rigorous method to obtain an upper bound is the application of the triangle inequality to the sum in (3). Unfortunately, it turns out that this bound is by far too large and the computational effort is drastically high, since a vast majority of points is rejected in the acceptance-rejection method (e. g.,

204 evaluations of the KDE were needed for one sample in numerical experiments). In the numerical experiments presented in this paper, the upper bound was obtained by evaluation of the KDE on a equiangular grid with 10 000 points on the sphere and multiplication of the result by 1.1 to bound maxima of the KDE, which may occur between the grid points. This leads to an average of 2.4 evaluations of the KDE to obtain one sample in our numerical experiments, which improves the computational effort of the acceptance-rejection method in this case very much.

Note that for a single evaluation of the KDE in (3), the kernel has to be evaluated N times, which affects the practicability of KDEs massively if N is large. Hence, since $N \approx 10^7$ is realistic in our application, the use of KDEs leads to very high computational costs. In particular, in our numerical tests, approximately 2.4×10^7 evaluations of the Abel-Poisson kernel were needed to obtain one single sample Y_j in step 2 of Algorithm 1. Additionally, by using the Abel-Poisson kernel, we already have a kernel at hand that admits a closed representation, whereas other kernels on the sphere (for examples, see [15, Chapter 7]) are given as a Legendre series and do not permit such a representation. This increases the computational effort of this method even further, if different kernels are chosen. Since a real nonwoven fabric can consist of hundreds of fibers and, in principle, the fibers can be arbitrarily long, we face a big problem in the practical realization of the method. Specific numbers for computation times in comparison to the new method, which is developed in the following section, can be found in Table 1 later in this article.

To reduce the number of kernel evaluations, we consider the concept of sparsity as studied, for example, in the field of compressive sensing (cf. [12, Chapter 2]). This means, in our case, that the aim is to estimate the PDF by a linear combination

$$\sum_{n=1}^N \alpha_n K_\lambda(X_n \cdot \xi)$$

of kernels, where most of the coefficients α_n are zero, or at least, very small. Although the obtained result is technically not a kernel density estimator, we will call such a representation a *sparse kernel density estimator*. Note that this representation shows certain similarities to the approximation of functions in Sobolev spaces on the sphere with spherical splines (see [13]). In the literature, strategies to achieve a sparse representation of a function include optimization methods, greedy methods, and thresholding-based methods (cf. [12, Chapter 3]). Sparse KDEs based on optimization methods were studied in [2, 6, 23]. The arising optimization problems for the N unknowns α_n may also be difficult to handle in practice if N is large as in the presented application. Greedy methods have only rarely been used in the estimation of PDFs so far: a method for the parametric case with Gaussian kernels on \mathbb{R} is considered in [42], which can obviously not be applied to the presented problem, since no parametrization of the PDF is assumed to be known. The non-parametric case is the subject of [39] with an approach based on the linearization and minimization of a so-called negative log-likelihood loss function. Although the minimization of this specific loss function is a common method in statistics, apart from the linearization, it brings with it the difficulty that optimization methods have to be used in every iteration of the algorithm. Furthermore, in that approach parts of the loss function have to be recalculated in every iteration, which demands a high computational effort for a large number of points. These two difficulties do

not arise with the new algorithm presented below, which can be interpreted as a minimization of an L^2 -error.

4. ANALYSIS OF DIRECTIONS BY MEANS OF A GREEDY APPROXIMATION

Greedy algorithms have been a successful tool in the sparse approximation of functions in Banach and Hilbert spaces (for a survey, see [41]). Several variants of these algorithms have also been used in signal processing as *matching pursuits* (cf. [29]) and in the solution of inverse problems as *regularized functional matching pursuits* and its variants (cf. [9, 34, 35]). Note that greedy algorithms have also been used in statistics before, but usually for regression problems (cf. [3, 16]). In [17], density estimation in \mathbb{R}^d was considered, relying heavily on the Euclidean structure.

It is the aim of this section to transfer the basic concept of greedy algorithms to the estimation of PDFs. The classical (or *pure*) greedy algorithm for the approximation of functions in Hilbert spaces reads as follows (cf. [41, Chapter 2]).

Algorithm 2 (Pure Greedy Algorithm, PGA). *Let \mathcal{H} be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$. Choose a so-called dictionary $\mathcal{D} \subseteq \{g \in \mathcal{H} \mid \|g\| = 1\}$. To approximate an element $f \in \mathcal{H}$, iteratively generate a sequence of approximations $(f_k)_{k \in \mathbb{N}_0}$ in the following way:*

- (1) *Set $k := 0$, choose an initial approximation $f_0 \in \mathcal{H}$.*
- (2) *Find a dictionary element $d_{k+1} \in \mathcal{D}$ whose projection onto the subspace of \mathcal{H} , which is spanned by the current residual $r_k := f - f_k$, is maximal, in other words*

$$d_{k+1} = \operatorname{argmax}_{d \in \mathcal{D}} |\langle r_k, d \rangle|. \quad (5)$$

- (3) *Set*

$$\alpha_{k+1} := \langle r_k, d_{k+1} \rangle. \quad (6)$$

- (4) *Set $f_{k+1} := f_k + \alpha_{k+1} d_{k+1}$.*
- (5) *If a suitable stopping criterion is fulfilled: stop.*
Otherwise: increase k by 1 and return to step 2.

The approximation after K steps is consequently given as $f_K = f_0 + \sum_{k=1}^K \alpha_k d_k$.

Note that we use $\mathcal{H} = L^2(\mathbb{S}^{d-1})$ from now on.

The difficulty in the implementation of greedy algorithms in the application considered here lies in the computation of the inner products $\langle r_k, d \rangle = \langle f, d \rangle - \langle f_k, d \rangle$ for $d \in \mathcal{D}$ in (5) and (6), especially in the inner product $\langle f, d \rangle$, whereas the inner product $\langle f_k, d \rangle$ can be easily computed if the inner products $\langle d, d' \rangle$ and $\langle f_0, d \rangle$ are known for all $d, d' \in \mathcal{D}$. The reason for this difficulty is that f is unknown and that neither values of f nor values of functionals evaluated at f are given, which could be handled by a matching pursuit and a regularized functional matching pursuit, respectively. Instead, realizations of the random variable X are given, whose distribution possesses the density f .

To overcome this difficulty, we observe that by the definition of the inner product and the expectation value of a random variable, we have

$$\langle f, d \rangle = \int_{\mathbb{S}^{d-1}} d(\xi) f(\xi) \, d\omega(\xi) = \mathbb{E}_f [d(X)],$$

since the distribution of X possesses the density f . Moreover, we use the strong law of large numbers (see, e. g., [27, Chapter 5.3]) to obtain the approximation

$$\langle f, d \rangle = \mathbb{E}_f [d(X)] \approx \frac{1}{N} \sum_{n=1}^N d(X_n), \quad (7)$$

which we insert into (5) and (6) to obtain the following algorithm.

Algorithm 3 (Greedy algorithm for the estimation of PDFs). *Let $f \in L^2(\mathbb{S}^{d-1})$ be an unknown PDF and $X_1, \dots, X_N \in \mathbb{S}^{d-1}$ be a realization of a random variable X , whose distribution possesses the density f . Let a dictionary \mathcal{D} be given as in Algorithm 2.*

Generate a sequence $(f_k)_{k \in \mathbb{N}_0}$ of approximations of f iteratively according to the following scheme:

- (1) *Set $k := 0$, choose an initial approximation $f_0 \in L^2(\mathbb{S}^{d-1})$.*
- (2) *Find a dictionary element $d_{k+1} \in \mathcal{D}$ fulfilling the maximization property*

$$d_{k+1} = \operatorname{argmax}_{d \in \mathcal{D}} \left| \frac{1}{N} \sum_{n=1}^N d(X_n) - \langle f_k, d \rangle \right|. \quad (8)$$

- (3) *Set*

$$\alpha_{k+1} := \frac{1}{N} \sum_{n=1}^N d_{k+1}(X_n) - \langle f_k, d_{k+1} \rangle. \quad (9)$$

- (4) *Set $f_{k+1} := f_k + \alpha_{k+1} d_{k+1}$.*
- (5) *If a suitable stopping criterion is fulfilled: stop.*
Otherwise: increase k by 1 and return to step 2.

Note that one would expect, from the theoretical point of view, a condition on the dictionary like $\overline{\operatorname{span} \mathcal{D}} = \mathcal{H}$. However, from the computational point of view this condition can never be fulfilled, unless \mathcal{H} is finite-dimensional, since \mathcal{D} would have to be comprised of infinitely many elements, which cannot be realized on a computer. The maximization in (8) can consequently be accomplished by evaluating the term that has to be maximized for every dictionary element and choosing the dictionary element with the maximal value. Note also that the maximizer in (8) does not need to be unique. In this case, we choose one arbitrary maximizer among all available maximizers.

By our motivation, the dictionary will consist of the kernels arising in the KDE and is consequently given as

$$\mathcal{D} := \{ \xi \mapsto K(X_n \cdot \xi) \mid n = 1, \dots, N \}$$

for a given kernel K .

In this case, both in (8) and (9), the kernel has to be evaluated at least $(N+1)N/2$ times, which—at the first sight—brings with it the same difficulties already discussed in Section 3 if N is large. Two arguments can be given which reduce this difficulty in practice:

First, the greedy algorithm has to be applied only once per data set to obtain a sparse estimation of the PDF. The coefficients α_k and the dictionary elements chosen by the greedy algorithm can be stored and reused, when an estimation of the PDF is needed in other algorithms. On the contrary, when using a KDE itself, the kernel has to be evaluated very often inside these other algorithms. This fact makes

it possible to execute the greedy algorithm on a powerful and fast computer system, for example a large cluster, to obtain a list of chosen coefficients and kernels, and subsequently use a slower computer, for example an ordinary desktop computer, for the further applications.

Secondly, the term

$$\frac{1}{N} \sum_{n=1}^N d(X_n) \quad (10)$$

arising in (8) for every $d \in \mathcal{D}$, which represents an empirical expectation value, does not change in the course of the iteration. That is to say, (10) has to be computed and stored only once in the beginning of the algorithm and does not need to be recomputed in every iteration. The value computed prior to the iteration can then also be used in (9). Additionally, the value of (10) for some d can be computed independently from the other dictionary elements. Thus, this process is ideal for parallelization, which improves the performance on those fast computer systems mentioned before.

Note that, in principle, this method can be extended to other types of dictionary elements apart from kernels, for example, spherical harmonics or wavelets, which have not been implemented for the numerical experiments in this paper. It would also be possible to use kernels with different concentration parameters for the dictionary. The use of such overcomplete dictionaries turned out to be useful for spherical inverse problems in the geosciences (see [4, 8, 9, 10, 11, 33, 34, 35, 40]). Further note that greedy algorithms with dictionaries consisting of kernels also appear in a method called *kernel matching pursuit* in the field of machine learning (cf. [43]).

5. NUMERICAL EXPERIMENTS

The Abel-Poisson kernel as introduced in (4) was chosen with a parameter $h = 0.9$ for all of the following numerical experiments. Since Algorithm 3 uses the approximation in (7), and convergence for the PGA is only known if the inner product can be evaluated exactly, a synthetic example is presented first in this section to empirically study the convergence of the algorithm. We proceed by a more detailed description of the CT data set used for the following computations. We continue by presenting simulation results using the greedy approximation of the PDF and comparing computation times between the use of KDEs and the greedy approximation. Finally, we conclude this section by presenting some first results about the validation of fiber laydown models. For this purpose, we also use the greedy algorithm.

5.1. A synthetic example. To create a synthetic example to test the convergence of the greedy algorithm, we choose the PDF $f(\xi) := Q_{0.6}^{(3)}(\xi \cdot \varepsilon^3)$, where the Abel-Poisson kernel is concentrated around $\varepsilon^3 = (0, 0, 1)^T$ and apply the acceptance-rejection method (cf. section 3 of this paper) to sample a data set X_1, \dots, X_N from this PDF, where $N = 10^6$. The PDF and 2000 points from the data set are presented in Figure 4.

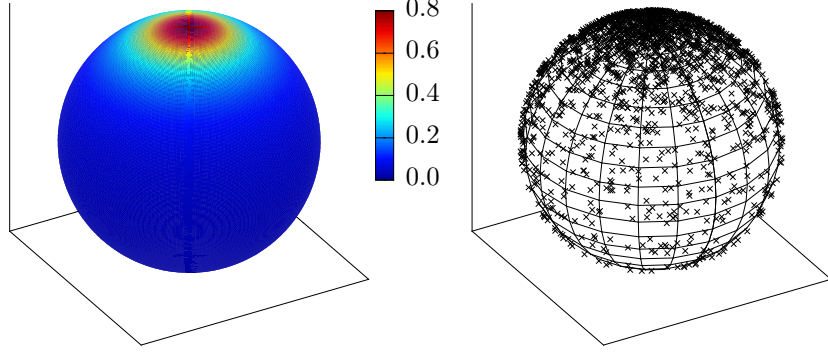


FIGURE 4. Left: PDF from the synthetic example, an Abel-Poisson kernel with parameter $h = 0.6$, centered at $\varepsilon^3 = (0, 0, 1)^T$; Right: 2000 of in total 10^6 data points sampled from that PDF with the acceptance-rejection method.

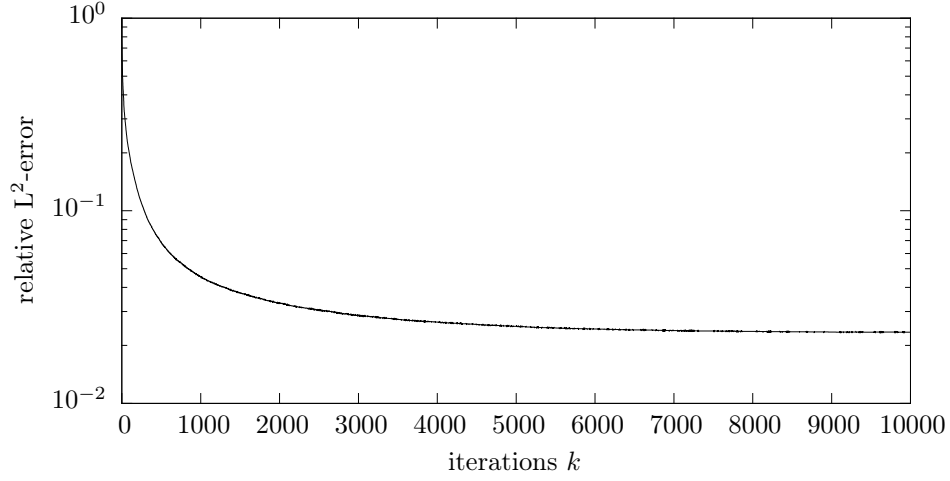


FIGURE 5. Semi-logarithmic plot of the relative $L^2(\mathbb{S}^{d-1})$ -error $\|f - f_k\|/\|f\|$ of the first 10 000 iterations of the greedy algorithm, when applied to the synthetic example.

In this synthetic case, where the PDF is already known, it is possible to compute the relative $L^2(\mathbb{S}^{d-1})$ -error $\|f - f_k\|/\|f\|$ explicitly due to properties of the Abel-Poisson kernel. A semi-logarithmic plot of this relative error for 10 000 iterations of the greedy algorithm can be found in Figure 5.

The algorithm shows a convergent behavior, at least in these first 10 000 iterations. Multiple performed experiments show that the number of data points influences the convergence of the algorithm in the following way: the more iterations are performed, the more data points are needed such that the algorithm shows

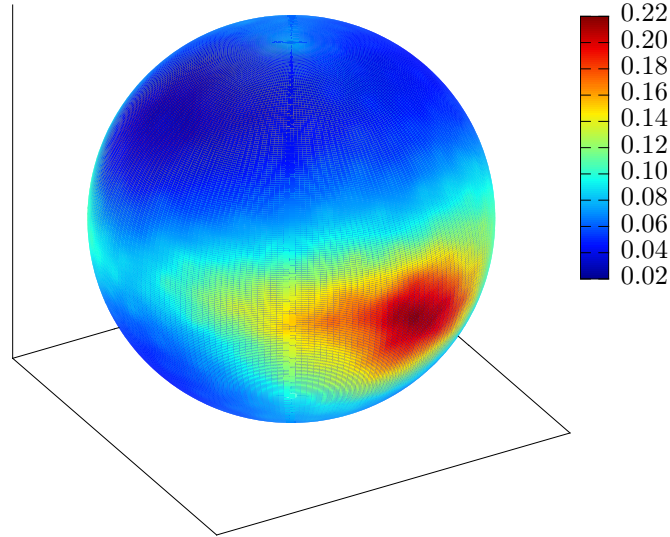


FIGURE 6. Sparse estimation of the PDF of a real data set after 10 000 iterations of the greedy algorithm.

a convergent behavior. This is plausible, since, having reached a certain level of accuracy, the approximation can only be improved with additional information on the function to be approximated. The dependence of the convergence on the number of data points is a subject of future research.

5.2. Description of the CT data set. We deal with a spunbond process of the industrial company Oerlikon Neumag. After specifying several production parameters, the company produced samples of nonwovens which will be used for the presented numerical experiments. The samples were analyzed by the department *Image Processing* of the Fraunhofer ITWM with its 3D-microtomography scanner. The CT scan of the real samples delivers a real-valued third-order tensor with so-called gray values. The gray values are mapped to local fiber orientations at each voxel with the help of an eigenvalue analysis of the Hessian matrix of the second partial derivatives of the gray values. For more details, the reader is referred to [37]. As already mentioned in the introduction, the microtomography scanner possesses a resolution of $1\text{ }\mu\text{m}$ to $70\text{ }\mu\text{m}$ while the sample size is 1 mm^3 to 10 mm^3 . For the data set that we use, this high resolution leads to $N = 9\,600\,558$ points on \mathbb{S}^2 , which will be the basis for our considerations.

5.3. Application of the greedy and simulation algorithms. We applied Algorithm 3 on the CT data set which was described in the previous section. The result is a sparse kernel density estimator, which is depicted in Figure 6.

Since the error of the estimation can no longer be explicitly computed, in Figure 7 the absolute values of the chosen coefficients $|\alpha_k|$ are plotted, to get an impression of the size of the corrections which the greedy algorithm performs in each of the first 10 000 iterations. This indicator for the size of corrections also shows a convergent

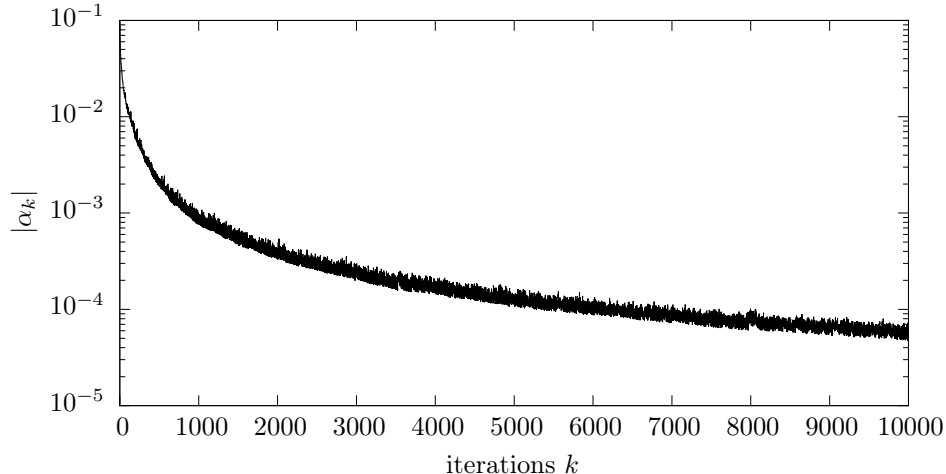


FIGURE 7. Absolute values of the chosen coefficients in the first 10 000 iterations of the greedy algorithm, applied to the real data set.

behavior, similarly to the synthetic example above.

Note that the KDE in Figure 3 and the sparse estimator in Figure 6 show a clear qualitative similarity. Quantitatively, one recognizes certain differences both in the values of the estimator (the maximum is 0.18 for the KDE and 0.22 for the greedy approximation) and in the structure of the estimator (the structures are rougher in the result of the greedy approximation). This is no contradiction, since the greedy approximation and the KDE are both estimations of the PDF, but the greedy approximation is no approximation of the KDE. Thus, depending on the choice of parameters h both for the KDE and the dictionary of the greedy algorithm, both estimations can be different while still estimating the same PDF.

Remember that our initial motivation for the development of a greedy algorithm was the inefficiency of Algorithm 1 if the PDF is estimated by a kernel density estimator. For this reason, we use the approximation $f_{10\,000}$ (generated by the greedy algorithm above) for a 3D simulation of nonwoven fabrics. A discretized fiber that has been generated this way is shown in Figure 8.

A comparison of computation times of the simulation algorithm is shown in Table 1.

In this table, we compare the uses of the KDE and the sparse KDE as well as the application of two different upper bounds for the PDF in the acceptance-rejection method (see the considerations in section 3). The following raw data are listed in the table: the number of samples X_j that have been generated, the number of evaluations of the estimated PDF that were needed to generate these samples, and the CPU time that was consumed. Note that there is a difference of the number of samples between the four presented scenarios due to the high computational effort that would be needed for the more inefficient methods. For that reason, to achieve a better comparison, we add the average number of evaluations of the estimated PDF per sample, the average CPU time per sample, and the time that would be needed to simulate a nonwoven fabric with 100 fibers with 100 000 segments each, to the table. One can see that the use of the greedy approximation makes

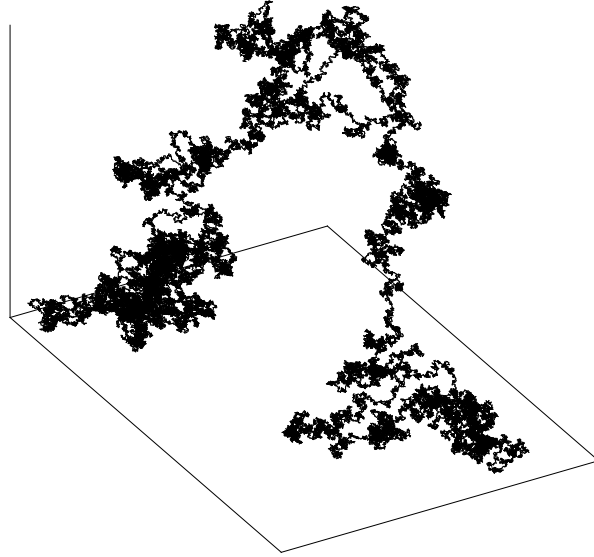


FIGURE 8. Fiber, consisting of 100 000 points, simulated by Algorithm 1 in conjunction with a greedy approximation of the PDF.

TABLE 1. Computation times for different estimates of the PDF and different upper bounds for the estimated PDF. For the comparison “CPU time/nonwoven”, it has been assumed that a simulated nonwoven consists of 100 fibers with 100 000 segments each. The presented data show an enormous saving of computation time when using the newly developed greedy algorithm.

upper bound	KDE		Greedy approximation	
	triangle ineq.	evaluation	triangle ineq.	evaluation
samples	420	25 000	100 000	100 000
evaluations	85 607	59 894	258 022	166 128
CPU time	16 022 s	8637.3 s	37.543 s	24.903 s
eval./sample	204	2.40	2.58	1.66
CPU time/sample	3.81×10^1 s	3.45×10^{-1} s	3.75×10^{-4} s	2.49×10^{-4} s
CPU time/nonwoven	105 833 h	958 h	1.04 h	0.69 h

Algorithm 1 more efficient by multiple orders of magnitude. Moreover, it is clear that the upper bound that is determined by applying the triangle inequality is much worse for the KDE, since there are much more summands in the estimator, such that a comparison of these numbers is not fair. Nevertheless, if the more efficient variant by evaluation on a fine grid is used for both estimators, the results are comparable. In that case, the simulation time for a nonwoven with 100 fibers with 100 000 line segments each, is reduced from 958 h to 0.69 h, thus from nearly 40 days to 41 minutes, a factor of nearly 1400.

Unfortunately, as in the case of the other known simulation methods for nonwovens (cf. the introduction), it is not trivial to validate the output of the simulation algorithm. By construction of the method, the output will fulfill the validation criterion introduced in the following section. Nevertheless, further research should be conducted on the topic of evaluating the quality of the simulations.

5.4. Validation of fiber laydown models. Another problem that can be attacked by using the developed greedy algorithm is the validation of a fiber laydown model, which has been developed at Fraunhofer ITWM and which is implemented in their simulation tool FIDYST (see [19]). For this purpose, we analyze the same data set generated by CT scans of real nonwoven fabrics as above and the result of a FIDYST simulation using the greedy algorithm introduced before.

As already mentioned before, the company Oerlikon Neumag has specified certain production parameters, which were used for the fabrication of the samples. For example, the belt speed was set to 38 m/min, the mean spinning speed is 79.4 m/s, and the fibers are made of polypropylene with a fiber diameter of 1.2×10^{-5} m. All specified parameters are used as input to FIDYST, which simulates the air flow and the fiber dynamics of this process—until the filaments are laid down on the conveyor belt. We prescribed a total simulation time of 0.5 s. Due to the mean spinning speed, we end up with a final filament length of 39.7 m. For the upcoming analysis, we neglect the fiber parts in the air and restrict ourselves to the laid-down parts on the conveyor belt. With a spatial discretization of the fibers of 2.5×10^{-4} m this leads to approximately 150 000 points per simulated fiber.

Since FIDYST can only produce a two-dimensional output, we face the problem to compare a 2D data set from the simulation with a real 3D data generated by CT scans. Let $\mathbb{X} = \{X_1, \dots, X_N\} \subseteq \mathbb{S}^2$ denote the CT data set and $\mathbb{Y} = \{Y_1, \dots, Y_M\} \subseteq \mathbb{S}^1$ the FIDYST data set, where $N, M \in \mathbb{N}$ are the numbers of data points for both types of data. It is known from the measurement process, that the third component of the CT data set corresponds to the direction perpendicular to the moving belt in the production process. Consequently, we overcome the challenge of comparing data sets in different dimensions by generating a 2D data set $\tilde{\mathbb{X}} = \{\tilde{X}_1, \dots, \tilde{X}_N\} \subseteq \mathbb{S}^1$ by projection onto the first two components.

The data sets used in the following consist of $N = 9\,600\,558$ (CT data) and $M = 3\,137\,432$ (FIDYST data) points, where we concatenated data from 10 distinct simulations (all initialized with the same parameters), and symmetrized the data. The application of the algorithm on $\tilde{\mathbb{X}}$ and \mathbb{Y} yields the results depicted in Figure 9. A possible interpretation of Figure 9 is as follows: it can be seen that the distribution of filament directions of the simulated fiber possesses maxima and minima at the same angles as the real fibers. However, the variance of the directions is much larger in the CT data set than in the FIDYST data set. Multiple calculations show that this effect also arises with other process parameters. It seems to be the case that the model used by FIDYST could be improved to better match the real world data. Necessary modifications of the model are the subject of further research.

Note that, so far, there was no way to measure the consistency of the model with reality. For the first time, the introduced method offers a possibility to do so.

6. CONCLUSIONS

In this paper, we dealt with the three-dimensional simulation of nonwovens. Using data from computer tomography scans of real nonwoven fabrics, we proposed

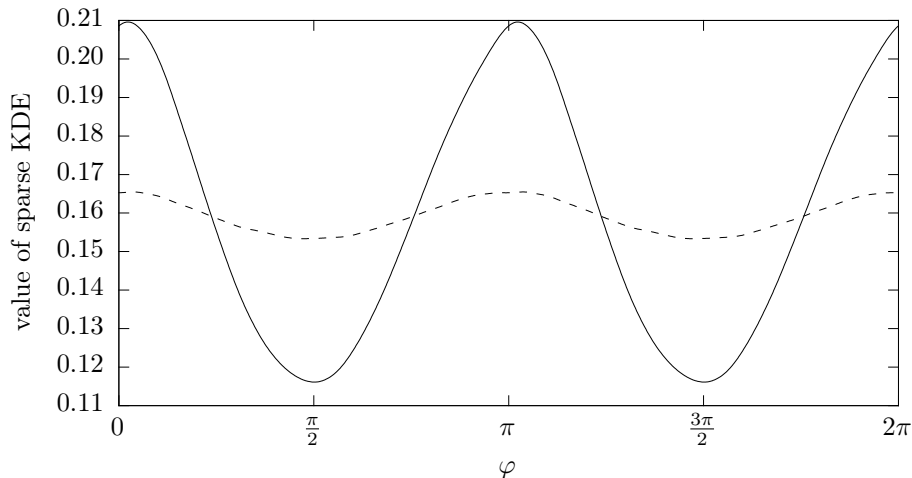


FIGURE 9. Sparse KDEs of the CT data set $\tilde{\mathbb{X}}$ (solid line) and the FIDYST data set \mathbb{Y} (dashed line). Two-dimensional Abel-Poisson kernels with $h = 0.9$ were used in both cases. The sphere \mathbb{S}^1 is identified with the interval $[0, 2\pi)$ on the horizontal axis.

an algorithm for the simulation of nonwovens. The basis of this algorithm is an estimate of the distribution of fiber directions inside the nonwoven, that is a probability density function. It turned out that using kernel density estimators, which are a standard technique in nonparametric statistics, leads to a very high computational effort when applying the simulation algorithm due to the high amount of available data. By introducing a greedy algorithm, which yields a sparse estimator of the unknown probability density, the computation time of the fiber simulation algorithm could be reduced by a factor of 1400, from more than a month to less than an hour.

The results presented in the work at hand give rise to several other topics that should be addressed in future research. First, the convergence of the greedy algorithm should be analyzed theoretically, in particular the dependence of the convergence on the number of data points. Secondly, in this paper, we restricted ourselves to a dictionary consisting of Abel-Poisson kernels centered at the data points in analogy to kernel density estimators. Of course, other functions could be used, for example Abel-Poisson kernels with different parameters h , that may also be centered at a regular grid instead of the data points. We could also add global functions like spherical harmonics to the dictionary. Thirdly, the presented simulation algorithm is not limited to the spunbond process. Several other production processes for nonwovens or also other technical textiles could be simulated with the given method, as long as directional CT data are available. Additionally, the simulation algorithm could be extended to include different belt movements than the one, which is already incorporated in the data set. Finally, the approach for the validation of fiber laydown models should be pursued further. The reason for the inconsistency of the distribution of fiber directions between the FIDYST model and the CT data should be analyzed. Also other fiber laydown models, like the stochastic surrogate model, could be analyzed with the given validation approach.

Furthermore, different models, like the FIDYST and the surrogate model, could be compared to each other, instead to the CT data, by our approach.

REFERENCES

1. W Albrecht, H Fuchs, and W Kittelmann (eds.), *Nonwoven Fabrics: Raw Materials, Manufacture, Applications, Characteristics, Testing Processes*, Wiley-VCH, Weinheim, 2003.
2. A Banerjee and P Burlina, *Efficient particle filtering via sparse kernel density estimation*, IEEE Trans. Image Process. **19** (2010), 2480–2490.
3. AR Barron, A Cohen, W Dahmen, and RA DeVore, *Approximation and learning by greedy algorithms*, Ann. Statist. **36** (2008), 64–94.
4. P Berkel, D Fischer, and V Michel, *Spline multiresolution and numerical results for joint gravitation and normal-mode inversion with an outlook on sparse regularisation*, Int. J. Geomath. **1** (2011), 167–204.
5. T Cacoullos, *Estimation of a multivariate density*, Ann. Inst. Statist. Math. **18** (1966), 179–189.
6. F Chen, H Yu, J Yao, and R Hu, *Robust sparse kernel density estimation by inducing randomness*, Pattern Anal. Appl. **18** (2013), 367–375.
7. L Devroye, *Non-Uniform Random Variate Generation*, Springer, New York, 1986.
8. D Fischer, *Sparse Regularization of a Joint Inversion of Gravitational Data and Normal Mode Anomalies*, Ph.D. thesis, Geomathematics Group, University of Siegen, 2011, published by Dr. Hut, München.
9. D Fischer and V Michel, *Sparse regularization of inverse gravimetry—case study: spatial and temporal mass variations in South America*, Inverse Problems **28** (2012), 065012.
10. ———, *Automatic best-basis selection for geophysical tomographic inverse problems*, Geophys. J. Int. **193** (2013), 1291–1299.
11. ———, *Inverting GRACE gravity data for local climate effects*, J. Geod. Sci. **3** (2013), 151–162.
12. S Foucart and H Rauhut, *A Mathematical Introduction to Compressive Sensing*, Birkhäuser, New York, 2013.
13. W Freeden, *On spherical spline interpolation and approximation*, Math. Method. Appl. Sci. **3** (1981), 551–575.
14. W Freeden and K Hesse, *On the multiscale solution of satellite problems by use of locally supported kernel functions corresponding to equidistributed data on spherical orbits*, Studia Sci. Math. Hungar. **39** (2002), 37–74.
15. W Freeden and M Schreiner, *Spherical Functions of Mathematical Geosciences: A Scalar, Vectorial, and Tensorial Setup*, Springer, Berlin, 2009.
16. JH Friedman and W Stuetzle, *Projection pursuit regression*, J. Amer. Statist. Assoc. **76** (1981), 817–823.
17. JH Friedman, W Stuetzle, and A Schroeder, *Projection pursuit density estimation*, J. Amer. Statist. Assoc. **79** (1984), 599–608.
18. M Galassi, J Davies, J Theiler, B Gough, G Jungman, P Alken, M Booth, and F Rossi, *GNU Scientific Library Reference Manual*, 3 ed., Network Theory, Bristol, 2009.
19. S Gramsch, D Hietel, and R Wegener, *Optimizing spunbond, meltblown, and airlay processes with FIDYST*, Melliand Int. **21** (2015), 115–117.
20. M Grothaus, A Klar, J Maringer, P Stilgenbauer, and R Wegener, *Application of a three-dimensional fiber lay-down model to non-woven production processes*, J. Math. Ind. **4:4** (2014).
21. T Götz, A Klar, N Marheineke, and R Wegener, *A stochastic model and associated Fokker-Planck equation for the fiber lay-down process in nonwoven production processes*, SIAM J. Appl. Math. **67** (2007), 1704–1717.
22. P Hall, GS Watson, and J Cabrera, *Kernel density estimation with spherical data*, Biometrika **74** (1987), 751–762.
23. X Hong, S Chen, and CJ Harris, *Sparse kernel density estimation technique based on zero-norm constraint*, The 2010 International Joint Conference on Neural Networks, IEEE, 2010, IJCNN, Barcelona, 18–23 July 2010.
24. MC Jones, JS Marron, and SJ Sheather, *A brief survey of bandwidth selection for density estimation*, J. Amer. Statist. Assoc. **91** (1996), 401–407.

25. A Klar, N Marheineke, and R Wegener, *Hierarchy of mathematical models for production processes of technical textiles*, ZAMM Z. Angew. Math. Mech. **89** (2009), 941–961.
26. A Klar, J Maringer, and R Wegener, *A 3d model for fiber lay-down in nonwoven production processes*, Math. Models Methods Appl. Sci. **22** (2012), 1250020.
27. A Klenke, *Probability Theory*, Springer, London, 2014.
28. P Kloeden and E Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, Berlin, 1992.
29. SG Mallat and Z Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Process. **41** (1993), 3397–3415.
30. N Mao and SJ Russell, *Modelling of nonwoven materials*, Modelling and predicting textile behaviour (X Chen, ed.), Woodhead, Cambridge, 2010, pp. 180–224.
31. N Marheineke and R Wegener, *Modeling and application of a stochastic drag for fiber dynamics in turbulent flows*, Int. J. Multiphase Flow **37** (2011), 136–148.
32. V Michel, *Lectures on Constructive Approximation*, Birkhäuser, Basel, 2013.
33. ———, *RFMP: An iterative best basis algorithm for inverse problems in the geosciences*, Handbook of Geomathematics (W Freeden, MZ Nashed, and T Sonar, eds.), Springer, Berlin, 2 ed., 2015, pp. 2121–2147.
34. V Michel and R Telschow, *A non-linear approximation method on the sphere*, Int. J. Geomath. **5** (2014), 195–224.
35. ———, *The regularized orthogonal functional matching pursuit for ill-posed inverse problems*, SIAM J. Numer. Anal. **54** (2016), 262–287.
36. E Parzen, *On estimation of a probability density function and mode*, Ann. Math. Stat. **33** (1962), 1065–1076.
37. C Redenbach, A Rack, K Schladitz, O Wirjadi, and M Godehardt, *Beyond imaging: on the quantitative analysis of tomographic volume data*, Int. J. Mater. Res. **103** (2012), 217–227.
38. M Rosenblatt, *Remarks on some nonparametric estimates of a density function*, Ann. Math. Stat. **27** (1956), 832–837.
39. S Rosset and E Segal, *Boosting density estimation*, Advances in Neural Information Processing Systems 15 (S Becker, S Thrun, and K Obermayer, eds.), MIT Press, 2003, NIPS, Vancouver, 9–14 December 2002, pp. 657–664.
40. R Telschow, *An Orthogonal Matching Pursuit for the Regularization of Spherical Inverse Problems*, Ph.D. thesis, Geomathematics Group, University of Siegen, 2014, published by Dr. Hut, München.
41. VN Temlyakov, *Greedy approximation*, Acta Numer. **17** (2008), 235–409.
42. JJ Verbeek, N Vlassis, and B Kröse, *Efficient greedy learning of gaussian mixture models*, Neural Comput. **15** (2003), 469–485.
43. P Vincent and Y Bengio, *Kernel matching pursuit*, Mach. Learn. **48** (2002), 165–187.
44. R Wegener, N Marheineke, and D Hietel, *Virtual production of filaments and fleeces*, Currents in Industrial Mathematics (H Neunzert and D Prätzel-Wolters, eds.), Springer, Berlin, 2015, pp. 103–162.